

점진적 동작

유관희*, 고인영*, 김일수*, 하중성**

*충북대학교 정보산업공학과 및 컴퓨터교육과

**우석대학교 컴퓨터공학과

e-mail:khyoo@cbucc.chungbuk.ac.kr

Progressive Motions

Kwan-Hee Yoo*, In-Young Ko*, Il-Soo Kim*
and Jong-Sung Ha**

*Dept of Information Industrial Engineering and Dept. of
Computer Education, Chungbuk National University

**Dept. of Computer Engineering, Woosuk University

요 약

모션 캡처 장비 혹은 모션 생성 소프트웨어에 의해 만들어진 캐릭터 모션이 3차원 게임 및 애니메이션에서 캐릭터의 동작을 시뮬레이션하기 위해 사용된다. 그러나 생성된 캐릭터 모션이 세밀하면 세밀할수록 캐릭터의 동작은 자연스럽게만 그래픽스 시스템에서 렌더링 성능, 전송 대역폭 및 저장 능력에 문제를 발생시킨다. 본 논문에서는 임의의 캐릭터 동작을 효율적으로 저장하고 전송할 수 있는 점진적 동작(progressive motions) 기법을 제시한다. 점진적 동작 기법은 동작의 단순화 및 동작의 LOD(level of detail)에 효과적으로 적용될 수 있다.

1. 서론

최근들어 모션 캡처 장비 혹은 모션 생성 소프트웨어에 의해 만들어진 캐릭터 모션이 3차원 온라인 게임과 3차원 애니메이션에 다양하게 사용되고 있다. 이렇게 만들어진 캐릭터 모션들에 대해 자연스러운 동작을 생성하기 위해서는 대략 초당 30 프레임의 지원을 받도록 되어 있다. 따라서 30 분짜리 캐릭터 동작을 만들기 위해서는 30분 * 60초/분 * 30 프레임/초 = 54,000 동작이 생성되어야 할뿐만 아니라 이들 동작이 화면에 출력되기 위해서는 그래픽스 시스템의 렌더링 과정을 거쳐야 한다. 따라서 시점을 고려하지 않고 모든 캐릭터에 대해 해당 동작을 모두 렌더링 파이프라인에 보낸다면 렌더링 성능에 문제가 있을 수 있다.

이러한 문제점을 제거하기 위해 모션에 대한 LOD(level of detail)에 관한 연구가 진행되어 왔다. Lee 등[1]은 다양한 모션 편집 작업을 쉽게 처리하

기 위해 다해상도(multiresolution) 기법을 제시하였고 이 기법을 모션 LOD에 적용하였다. 모션 동작의 부드러운 측면에서 상당히 좋은 결과를 제시하고 있다. 또 다른 접근 방법으로 캐릭터를 구성하는 뼈대(skeleton)의 단순화로 Ercegovac 등[4]은 시점에 따라 뼈대를 어떻게 단순화할 것인지에 대한 방법을 제시하였으며, 단순화한 후 역운동학 및 동작재설정 기법을 이용하여 동작을 자연스럽게 하는 방법을 제시하였다. 이들 방법과는 달리 캐릭터 동작을 시뮬레이션하는 과정에서 시점에 따라 동작 시뮬레이션 하는 기법을 어떻게 단순화할 것인지에 대한 연구가 있었다[2,3]. 이러한 동작 단순화 연구 결과는 다양한 분야에 응용될 수 있으며, 특히 수백 명에서 수천 명으로 구성된 군중의 동작을 효율적으로 시뮬레이션하기 위해서는 렌더링의 성능이 매우 중요해 동작의 LOD 기능이 필수적으로 요구되고 있다[7].

본 논문에서는 점진적 매쉬 기법[8]을 변형한 동

작의 LOD 등을 효과적으로 제공할 수 있는 점진적 동작(Progressive Motions) 기법을 제안한다. 모션 캡처 혹은 동작 생성 소프트웨어에 의해 만들어진 캐릭터 동작 집합이 있다고 가정할 때 점진적 동작 기법에서는 동작 병합(motion collapse)과 동작 분해(motion split)를 통해 동작의 LOD를 제공한다. 만약 특정 캐릭터가 시점으로부터 멀어지면 그 캐릭터의 동작들을 병합하여 새로운 동작을 만들면서 전체 동작 프레임 줄여 나가고 반면 시점으로부터 캐릭터가 가까워지면 병합된 동작을 분해하여 좀더 세밀한 동작의 집합으로 표현한다. 이들 두 기법을 시점에 따라 적절하게 적용함으로써 렌더링 부담을 줄일 수 있다. 본 논문에서는 이들 두 명령을 어떻게 처리하는지에 대해 논의하며 또한 동작 병합시 자연스러운 동작을 생성하는 모션 모핑(motion morph) 기법을 제시한다.

본 논문의 구성은 제 2장에서는 점진적 동작 기법을 자세히 설명하고 제 3장에서는 구현 결과를 제시하며 마지막으로 연구 결과와 향후 연구 과제를 언급한다.

2. 점진적 동작

이번 장에서는 본 논문에서 제안하는 점진적 동작(progressive motions)을 처리하기 위한 요구되는 동작 병합(motions collapse)과 동작 분리(motions split)에 관해 논의하며, 동작 병합시 동작의 자연스러운 생성을 위해 동작 모핑(MotionMorph) 기법을 소개한다.

2.1 동작 병합 및 분해

이번 절에서는 캐릭터의 초기 동작을 보다 더 단순한 동작으로 만들 수 있는 기법인 동작 병합(motion collapse)을 먼저 소개하고 병합된 동작을 다시 분리하는 동작 분해(motion split)를 소개한다.

n 개의 프레임으로 구성된 캐릭터의 동작을 고려해 보자. 설명을 쉽게 하기 위해 i 번째 프레임에 대한 계층적으로 표현된 캐릭터 모션 상태를 X_i 라 표기한다. X_i 는 캐릭터 루트 위치 벡터 $P_{i,0}$ 와 가중치 루트 회전 값 $Q_{i,0}$ 그리고 루트로부터 계층적으로 이루어진 모든 몸체 k 에 대해 그의 부모에 관한 가중치 회전 값 $Q_{i,k}$ 으로 표기되며 식으로 표현하면 식 (1)과 같다.

$$X_i = [P_{i,0} \ Q_{i,0} \ Q_{i,1} \dots \ Q_{i,m}]^T \quad (1)$$

식 (1)에서 m 은 캐릭터를 구성하는 뼈대의 개수를 의미한다. 따라서 n 개의 프레임으로 구성된 캐릭터의 모든 동작 M^n 은 식 (2)과 같이 표현된다.

$$M^n = [X_0 \ X_1 \ \dots \ X_n]^T \quad (2)$$

캐릭터의 초기 동작을 M^0 이라 한다. 이 상태에서 동작을 단순화하는 효과적인 방법인 모션 병합(motion collapse)은 연속적인 두 개의 모션을 하나의 모션으로 병합함을 의미한다. 모션 병합 $mcol(X_i, X_{i+1})$ 은 모션 상태 X_i 와 X_{i+1} 을 합하여 새로운 모션 X_i 를 생성한다. 초기 모션 상태 M^0 로부터 n 번의 연속적인 모션 병합 과정을 적용함으로써 가장 단순한 모션 상태 M^0 이 생성된다.

$$M^n \xrightarrow{mcol_{n-1}} M^{n-1} \xrightarrow{mcol_{n-2}} \dots \xrightarrow{mcol_1} M^1 \xrightarrow{mcol_0} M^0$$

위와 같은 모션 병합 과정에서 가장 중요한 요소가 어떤 기준으로 병합될 두 모션을 선택하는지에 관한 것이다. 본 논문에서는 특정 모션 상태 X_i 에서 다른 모션 상태 X_j 로 이동하기 위해 소요되는 비용함수 $D(X_i, X_j)$ 를 식 (3)과 같이 정의하여 사용하였다.

$$D(X_i, X_j) = wt \ T(X_i, X_j) + wr \ R(X_i, X_j) \quad (3)$$

$$T(X_i, X_j) = \| P_{i,0} - P_{j,0} \|$$

$$R(X_i, X_j) = \sum_{k=0}^m wr_k (\min(\| \log(Q_{i,k}^{-1} Q_{j,k}) \|, \| \log(Q_{i,k}^{-1} (-Q_{j,k})) \|))$$

식 (3)에서 wt 와 wr 는 각각 동작 병합을 위해 캐릭터의 전이 동작과 회전 동작이 얼마나 영향을 미치는지를 나타내는 가중치이며, wr_k 는 캐릭터의 k 번째 연결 부위의 회전 동작에 이 캐릭터에 얼마만큼의 영향을 미치는지를 나타내는 가중치이다. 분명 두 동작의 병합을 위해서는 식 (3)의 $D(X_i, X_j)$ 값이 최소화되는 i 프레임과 j 프레임을 선택하여야 한다. 그러나 i 프레임 측면에서 동작의 특성을 분석해보면 $D(X_i, X_j)$ 을 최소로 하는 j 프레임은 $i+1$ 프레임임을 알 수 있다. 그러므로 두 모션의 병합을 위해서는 먼저 식 (4)와 같은 i 번째 프레임을 선택하여야 한다.

$$\text{Minimize } 1 < i < n \ D(X_i, X_{i+1}) \quad (4)$$

식 (4)로부터 선택된 i 번째 프레임은 $(i+1)$ 번째 프레임과 병합이 이루어져 그 결과가 i 번째 프레임에 저장된다. 두 동작의 병합 과정은 제 2.2절에서 상세히 설명한다.

동작 분해 명령(msplit)은 두 동작 합성의 역 과정이므로 동작 합성에서와 같이 특정한 한 프레임을 연속된 두 프레임으로 분해한다. 따라서 동작 병합의 최종 상태인 M^0 으로부터 $n-1$ 번의 동작 분해 과정을 거치면 동작의 초기 상태인 M^n 가 얻어진다. 그 과정을 나타내면 다음과 같다.

$$M^0 \xrightarrow{msplit_0} M^1 \xrightarrow{msplit_1} \dots \xrightarrow{msplit_{n-2}} M^{n-1} \xrightarrow{msplit_{n-1}} M^n$$

동작 병합과 동작 분해가 서로 역 과정이므로 이를 효율적으로 처리하기 위해 본 논문에서는 병합 동작과 분해 동작을 각각 저장하는 두 개의 스택(stacks) Sc, Ss를 사용하였다. 동작 병합이 일어나는 두 프레임을 차례대로 Sc에 저장한다. 반복적으로 같은 작업이 매 두 동작의 합성이 처리될 때마다 처리된다. 동작 분해 명령이 요구되면 가장 최근에 병합된 위치를 찾은 후에 그 위치에 병합되어 저장된 모션 상태가 Ss에 저장되고 Sc에 저장된 두 동작이 그 위치에 있는 동작을 대치한다.

2.2 동작 모핑(MotionMorph)

식 (4)로부터 병합될 연속된 두 프레임이 선택되었다고 하자. 이들 두 동작과 이들 동작의 앞뒤 동작을 이용하여 새로운 동작을 생성하여야 한다. 분명 이 작업에서 가장 중요하게 고려되어야 할 사항이 어떻게 하면 부드러운 동작을 새롭게 생성하는가이다. 이를 위해 본 논문에서는 두 가지 변환, 즉 캐릭터의 루트 위치 구하기와 각 조인트 부분에서의 회전각 구하기를 처리한다.

먼저 캐릭터의 두 동작 X_i 와 X_{i-1} 을 병합하여 새로운 모션 상태 X'_i 를 생성한다고 가정해 보자. 즉 $X'_i = [P'_{i,0} \ Q'_{i,0} \ Q'_{i,1} \dots \ Q'_{i,m}]^T$ 을 구해보자. 이를 위해 먼저 캐릭터의 루트 위치인 $P'_{i,0}$ 을 구해야 한다. 본 논문에서는 네 개의 모션 상태 $X_{i-1}, X_i, X_{i+1}, X_{i+2}$ 의 각 루트의 위치 정보 $P_{i-1,0} \ P_{i,0} \ P_{i+1,0} \ P_{i+2,0}$ 을 보간하여 생성하는 Catmull-Rom 스플라인 커브를 이용하여 $P'_{i,0}$ 을 구하였다[5]. Catmull-Rom 스플라인 커브를 이용하여 $P'_{i,0}$ 을 구하는 구체적인 식은 식 (5)와 같다.

$$P'_{i,0} = P(u) = P_{i-1,0}(-su^3 + 2su^2 - su) + P_{i,0}(2.0-s)u^3 + (s-3)u^2 + 1 + P_{i+1,0}(s-2)u^3 + (3-2s)u^2 + su + P_{i+2,0}(su^3 - su^2) \quad (5)$$

본 논문에서는 식 (5)의 s 와 u 에 대해 각각 0.5를 주었다.

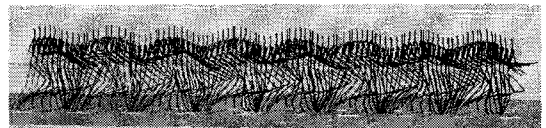
다음으로 캐릭터의 두 동작이 병합되어 새로운 동작을 생성시 캐릭터의 각 연결 부위에서의 새로운 회전각을 구해야 한다. 이를 위해 본 논문에서는 쿼터니언(quaternion)의 보간법으로 널리 사용되는 Shoemaker[6]가 제시한 Slerp을 식 (6)과 같이 이용하였다.

$$Q'_{ij} = \text{Slerp}(Q_{i,j}, Q_{i+1,j}, 0.5), j=0, \dots, m \quad (6)$$

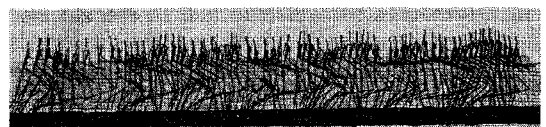
3. 구현 결과

이번 장에서는 달리는 캐릭터 동작에 대해 제 2장에서 제시한 점진적 동작 기법을 적용한 결과를 제시한다. 그림 1의 (a)가 101 프레임으로 구성된 달리는 캐릭터의 동작을 시뮬레이션한 결과이다. 그림 1의 (b)에서 (j)까지의 결과가 주어진 프레임에서 10 프레임씩을 동작을 병합할 때 생성된 결과를 보여주고 있다. 각 병합 단계별 결과를 살펴보면 캐릭터 루트의 전이 및 각 조인트 부분에서의 회전 변화가 그리 심하지 않는 부위에서 동작 병합이 이루어짐을 알 수 있다.

앞에서 설명한 바와 같이 동작의 분해는 동작 병합의 역과정이다. 따라서 그림 1의 (j)에서 10 프레임을 추가하면 (i)가 되며, (i)에 또 10 프레임을 추가하면 (h)가 된다. 이러한 작업을 반복적으로 적용하면 초기 동작의 집합을 나타내는 그림 1의 (a)가 생성된다.



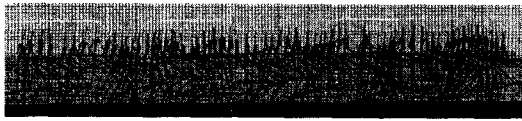
(a) 101 프레임



(b) 90 프레임



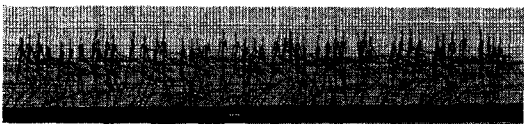
(c) 80 프레임



(d) 70 프레임



(e) 60 프레임



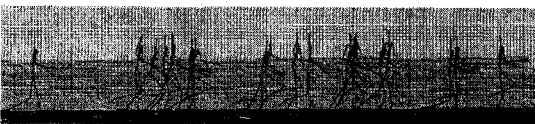
(f) 50 프레임



(g) 40 프레임



(h) 30 프레임



(i) 20 프레임



(j) 10 프레임

그림 1 점진적 동작 시뮬레이션 결과

4. 결론

본 논문에서는 Hoppe[8]가 제안한 점진적 메쉬 기술을 캐릭터의 모션에 적용한 점진적 동작 기법을 제시하였다. 제시된 결과는 시점이 캐릭터로부터 멀어지면 동작을 합성하여 보다 단순한 동작을 렌더링 하고 시점으로부터 가까워지면 동작을 분해하여 좀 더 세밀한 동작을 렌더링함으로써 3차원 게임 혹은 애니메이션에 효과적으로 적용될 수 있다.

향후 본 논문에서 제시한 점진적 동작 기법을 동

작 압축 및 시점에 따른 동작 단순화 및 점진적 동작 전송 기법에 어떻게 적용할 지에 대한 연구가 요구된다.

참고문헌

- [1] Jehee Lee and Sung Yong Shin, "Multiresolution Motion Analysis with Applications," *The international workshop on Human Modeling and Animation*, Seoul, pp. 131-143, June 2000.
- [2] D. A. Carson and J. K. Hodgins, "Simulation Level of Detail for Real-time Animation," *Graphics Interface*, 1997
- [3] Stephen Chenney, Jeffrey Ichnowski and D.A. Forsyth, "Dynamics Modeling and Culling," *IEEE Computer Graphics and Applications*, pp. 79-87, Mar. 1999
- [4] Vuk Ercegovac and Neoklis Polyzotis, *Motion Level of Detail, CS838 Final Report*, University of Wisconsin, 2001
- [5] D. Hearn and M.P. Baker, *Computer Graphics, Second Edition*, Prentice Hall International Editions, 1994
- [6] K. Shoemake, "Animating rotation with quaternion curves," *Computer Graphics(Proc. of SIGGRAPH 85)*, pp.245-254, 1985
- [7] D. Kim, H. Kim and S.Y. Shin, *Event Driven Crowd Simulation using Example Motions, Technical Report*, KAIST, 2002
- [8] H. Hoppe, "Progressive Meshes," *Computer Graphics(Proc. of SIGGRAPH 1995)*, pp.99-108, 1995